GENERATION OF FOURIER TRANSFORM HOLOGRAM ON DIGITAL COMPUTER

A thesis submitted to the Faculty of

Electrical Engineering

in partial fulfilment of the requirements of the

Degree of Master of Technology

Indian Institute of Technology

Kanpur

August, 1968

CERTIFICATE

Certified that this work on the "Generation of Fourier Transform Hologram on Digital Computer" has been carried out under my supervision and that this has not been submitted elsewhere for a degree.

[illegible]

Dr. B.Prasada

Professor

Department of Electrical Engineering

Indian Institute of Technology, Kanpur

August, 1968

ACKNOLEDGEMENTS

The author is deeply indebted to Dr. B. Prasada for his able guidance and encouragement throughout the course of this project. He also takes this opportunity to thank Dr. K. R. Sarma for making valuable suggestions.

ABSTRACT

A fast Fourier transform program for the generation and reconstruction of Fourier Transform Holograms is developed based on the algorithm suggested by Cooley & Tukey[*]. The theory underlying the Hologram generation & reconstruction and the algorithm for the fast Fourier transform are discussed. The savings in time and storage over the direct computational method are distinctly brought out. These savings are quite significant for large arrays. Certain applications of the FFT program in antenna pattern synthesis convolution integration etc. are also discussed.

Table of Contents

1. INTRODUCTION

Hologram techniques constitute a unique method of storing the three dimensional information (amplitude as well as phase) so that it can be recalledat will as a real image that can be inspected in detail.Hologram photography may help in solving complex problems in image formation such as character recognition,aberration balancing of lenses. Practical embodiments of these hologram applications have yet to be accomplished,since these are based on the use of coherent illumination.The feasibility studies can be carried out by simulating the optical process on a digital computer. The various steps involved in the generation & reconstruction of the Fourier Transform Hologram on a digital computer are (a) generation of redundant data points by interpolation (b) computation of the Fourier transform of the modified data (c) addition of a uniform light pattern with linearly increasing phase to the transform (d)computing the intensity of the complex signal to get the Fourier Transform Hologram (e) inverse Fourier transforming the hologram to get the actual data without the redundancy introduced.

Since the amount of data to be processed in image formation is usually prohibitively large from the viewpoint of computer storage,this calls for an efficient use of computer time & memory.The fast Fourier transform algorithm suggested by Cooley & Tukey[1] is one such method of efficient and economic computer usage.A Fortran IV program based on this algorithm has been written and debugged.The theory underlying Fourier Transform Holography,hologram generation and reconstruction technique,the details of the algorithm, the program and its application in antenna pattern synthesis convolution integration etc. are presented here.

2. GENERATION AND RECONSTRUCTION OF HOLOGRAMS

The magnitude and phase of a scattered wavefront can be recorded photographically by superimposing a coherent reference beam or background wave on the field striking the photographic plate.The recorded diffraction pattern,called a HOLOGRAM,bears little resemblence to the object,but contains most of the information required to reconstruct the object. The process has three essential parts:

(a) The defocussing or spatial frequency dispersion of the image.

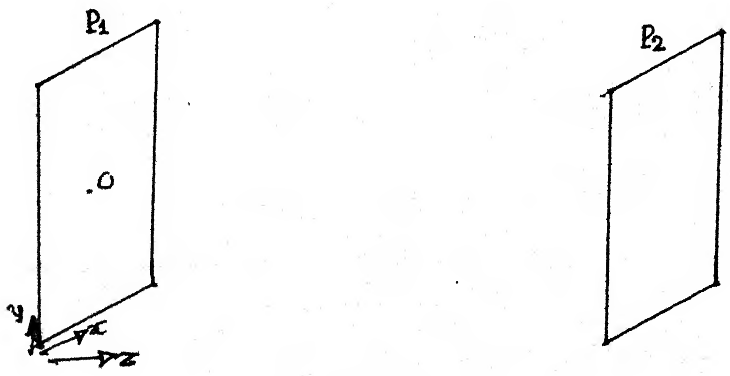
(b) The hologram recording process.

(c) Reconstruction.

2.i Dispersion Process:

Let a transparency having amplitude transmittance $s=s_b+s_r$ be placed at plane P_i.Amonochromatic plane wave illuminates the plane P_i and is modulated by the transparency,resulting in the light amplitude distribution

$$se^{j\omega t} = s_b+s_r(x,y)\ e^{j\omega t}$$



Construction of Hologram

At plane P_2, an "out of focus" image or Fresnel diffraction pattern of the transparency is formed. The image using the Fresnel-Kirchoff diffraction integral becomes:

$$X(x,y)=\frac{j}{\lambda z}\iint_{-\infty}^{\infty}\left[s_b+s_r(u,v)\right] e^{-j\frac{2\pi}{\lambda}\left[z^2+(x-u)^2+(y-v)^2\right]^{\frac{1}{2}}} dudv$$

Making the small angle approximation,

$$\left[z^2+(x-u)^2+(y-v)^2\right]^{\frac{1}{2}} \doteq z+\frac{(x-u)^2}{2z}+\frac{(y-v)^2}{2z}$$

$$X(x,y)=\frac{j}{\lambda z}\iint_{-\infty}^{\infty}\left[s_b+s_r(u,v)\right] e^{-j\frac{2\pi}{\lambda}\left[(x-u)^2+(y-v)^2\right]^{\frac{1}{2}}} dudv$$

where, a constant phase factor, $e^{-j\frac{2\pi}{\lambda}z}$, has been dropped.

$$X(x,y)=s_b+s_r(x,y)*f(x,y)$$

where

$$f(x,y)=\frac{j}{\lambda z}\, e^{-j\frac{2\pi}{\lambda}(x^2+y^2)}$$

$$X(x,y)=s_b+\vec{\mathcal{F}}\left[S_r(u,v).F(u,v)\right]$$

where

$$S_r(u,v)=\mathcal{F}\left[s_r(x,y)\right]$$

and $$F(u,v)=\mathcal{F}\left[f(x,y)\right]= e^{j\frac{2\pi}{\lambda}(u^2+v^2)}$$

2.2 <u>Recording</u>:

Since the photographic plate records only the absolute magnitude, therefore, the recorded hologram represents the function:

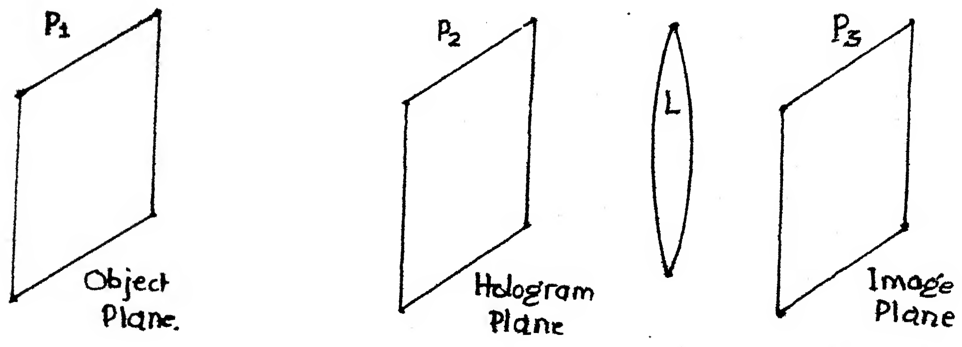
$$XX^*=s_b^{\ 2}+s_b(s_r*f)+s_b(s_r*f)^*+\left|s_r*f\right|^2 \qquad (1)$$

2.3 <u>Reconstruction</u>:

The reconstruction is carried out by illuminating the hologram with coherent light. Considering only the real image term of Eq. (1), $s_b(s_r*f)^*$, when this term is operated on by the dispersive filter, this becomes:

$$s_b(s_r*f)^{*}*f=\mathcal{F}\left[s_b.S_r^{\ *}(-u,-v)F^{*}(-u,-v).F(u,v)\right]$$
$$= s_b.s_r$$

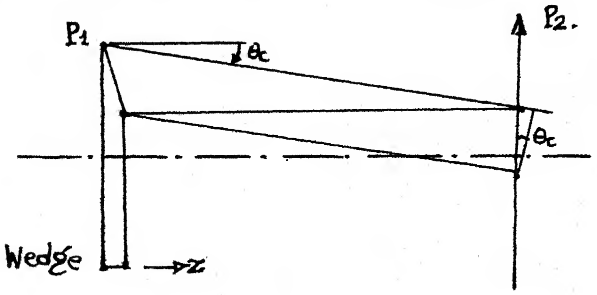
This term along with the bias term, s_b^2, contitutes the reconstructed image.



The object at plane P_i is imaged at plane P_3 and the hologram is recorded at some plane P_2 between P_i & P_3. When the hologram is produced, the original object is removed and the hologram is reinserted at P_2. The waves generated at P_2 are replicas of those that had impinged at P_2 when the hologram was made. These waves continue to P_3 and form an image.

If the phase function has the property, $F(u,v)=F^*(-u,-v)$, the light distribution at the hologram plane will be real and in construction there will be no interference from the twin-image.

The other method to eliminate twin-image



interference is to put the dispersed signal on a carrier as shown in the Fig, above. The dispersed signal can be represented by:

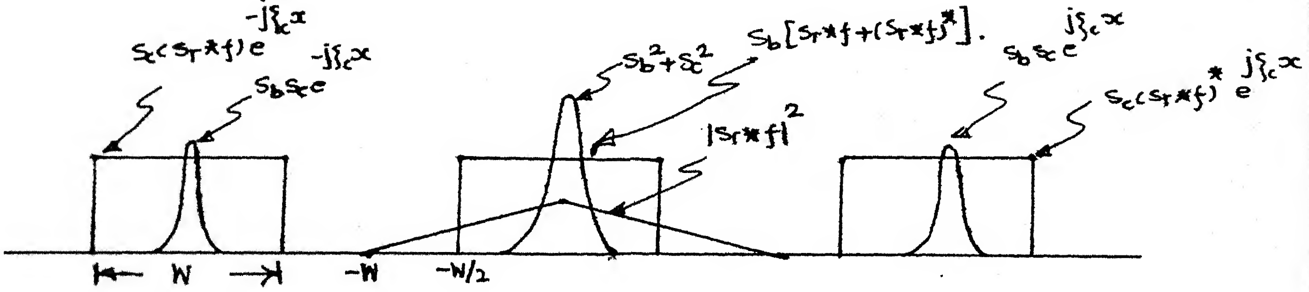
$$X=s_b+s_r*f+s_c e^{j\xi_c x}$$

where

$$\xi_c = \frac{2\pi\theta_c}{\lambda}$$

The recorded signal can be represented by:

$$XX^* = s_c^2 + s_b^2 + 2s_b \cdot s_c \cdot \cos \xi_c x + |s_r * f|^2 + s_b\left[s_r * f + (s_r * f)^*\right] + s_c\left[(s_r * f)e^{-j\xi_c x} + (s_r * f)^* e^{j\xi_c x}\right] \qquad (2)$$



$s_c(s_r * f)e^{-j\xi_c x}$
$s_b s_c e^{-j\xi_c x}$
$s_b^2 + s_c^2$
$s_b[s_r * f + (s_r * f)^*]$.
$s_b s_c e^{j\xi_c x}$
$s_c(s_r * f)^* e^{j\xi_c x}$
$|s_r * f|^2$
W
-W
-W/2

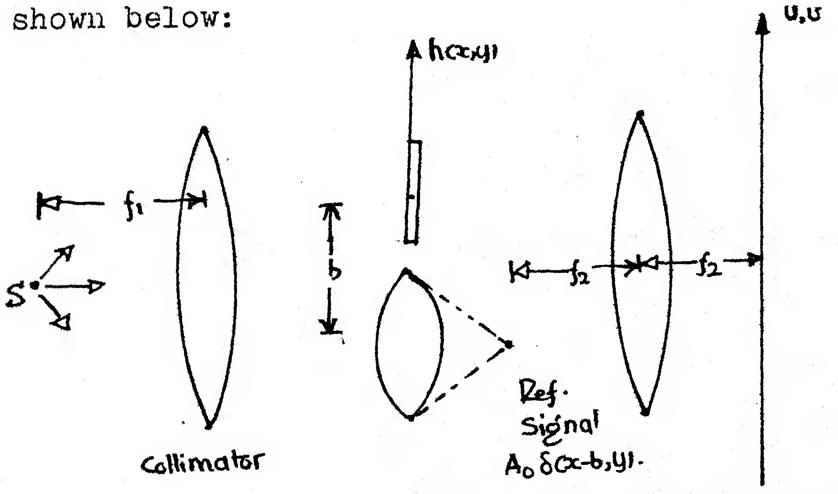
Spectrum of Hologram

If $s_r * f$ has bandwidth, W, the requirement that the spectra are non-overlapping is that

$$\xi_c \geq \frac{3}{2}W$$

2.4 Fourier Transform Holography:

For attaining high resolutions in wavefront reconstruction imaging, Fourier Transform Holography is employed. The recording procedure for such holography is shown below:



Let the reference signal be:

$$|A_o| e^{j\emptyset}$$

where

$\emptyset = bu+cv$

b=offset along the x-axis

c=offset along the y-axis

The complex amplitude at the focal plane of L_2 (by Fourier transformation) is given by:

$$A(u,v)=H(u,v)+P(u,v)$$

where

$$H(u,v)= |H| e^{j\theta} = \mathfrak{F}[h(x,y)]$$
$$P(u,v)= |A_o| e^{j\emptyset}$$

The intensity recorded on the plate is given by:

$$I(u,v)=A(u,v).A^*(u,v) = |H|^2 + |A_o|^2 + |H|\cdot|A_o| e^{j\theta} e^{-j\emptyset} + |H|\cdot|A_o| e^{-j\theta} e^{j\emptyset} \qquad (3)$$

The reconstruction is done by Fourier transformation of the recorded hologram, which gives:

$$R(x,y)=\text{reconstructed image} = |H|^2 + |A_o|^2 + |A_o|.h.\delta(x+b,y+c) + |A_o|.h.\delta(x-b,y-c) \qquad (4)$$

For negligible interference from the twin-image,

$$b,c \geq \frac{3}{2}W \qquad (5)$$

where WxW is the width of the original object.

3. COMPUTATIONAL STEPS INVOLVED IN THE GENERATION AND RECONSTRUCTION OF FOURIER TRANSFORM HOLOGRAMS

To generate & reconstruct the Fourier- Transform Hologram on a digital computer,the following steps are required:

(a) If the data has N1xN2 samples,redundancy is introduced by interpolation to generate 4N1xN2 samples since the hologram requires a minimum of four times as much band width as the signal to avoid twin-image interference.

(b) This data is Fourier transformed to get four times the number of Fourier coefficients required for the faithful reproduction of the original signal.

(c) The carrier signal having a constant amplitude and linearly increasing phase is added to the transformed array and the amplitude of the complex signal is computed to get the hologram

(d) To reconstruct the original signal,the hologram thus generated is Fourier inverse transformed.The inverse transform thus obtained contains the original signal,the twin-image,the auto-correlation function of the signal.Since the auto-correlation function of the signal requires twice as much band width as the signal; and the twin-image requires as much band width as the signal ,the total (minimum) band width needed for non overlapping spectra of the signal,the twin image,and the auto-correlation function is four times as large as the signal band width. Thus the redundant samples introduced minimize interference,and hence,the reproduced signal contains the actual time signal without any redundancy what so ever.

Since the amount of data to be handled is prohibitively large from the point of view of computer storage,this calles for an efficient use of computer time and storage.A multi-access,minimum time scheme is ideally suited for handling such large data arrays.The fast Fourier transform algorithm suggested by Cooley & Tukey is one such scheme.The theory underlying this algorithm is explained in the following sections.

4. FOURIER TRANSFORM REPRESENTATION OF DISCRETIZED PERIODIC SIGNALS

Let A(t) be a band limited sampled function of periodicity T, sampled at intervals of Δt, the Fourier transform and its inverse are then given by:

$$X(J)=\sum_{K=0}^{N-1} A(K)W^{JK} \qquad \text{for } J=0,1,2,\ldots\ldots\ldots,N-1$$

and

$$A(K)=\sum_{J=0}^{N-1} X(J)W^{-JK} \qquad \text{for } K=0,1,2,\ldots\ldots\ldots,N-1$$

where

$t=K.\Delta t$

$T=N.\Delta t$

$W=\exp(2\pi j/N)$

A derivation of this expression is given in Appendix-B

In the two dimensional case, the transform and its inverse can be written as:

$$X(J_1,J_2)=\sum_{K2=0}^{N2-1}\sum_{K1=0}^{N1-1} A(K_1,K_2)W1^{J1.K1}.W2^{J2.K2}$$

$$\text{for } J1=0,1,2,\ldots\ldots\ldots,N1-1$$
$$J2=0,1,2,\ldots\ldots\ldots,N2-1$$

$$A(K_1,K_2)=\sum_{J2=0}^{N2-1}\sum_{J1=0}^{N1-1} X(J_1,J_2)W1^{-J1.K1}.W2^{-J2.K2}$$

$$\text{for } K1=0,1,2,\ldots\ldots\ldots,N1-1$$
$$K2=0,1,2,\ldots\ldots\ldots,N2-1$$

where

$W1=\exp(2\pi j/N1)$

$W2=\exp(2\pi j/N2)$

5. COMPUTATIONAL TIME INVOLVED IN DIRECT CALCULATION OF FOURIER TRANSFORM

The number of operations in the one dimensional case is N^2, since from the very definition, N complex multiplications as well as additions are to be carried out for each point.

Similarly, in the two dimensional case there are Ni.N2 multiplications as well as additions for aech point and there are Ni.N2 points. Therefore, total number of operations to be carried out is given by:

$$(Ni.N2)^2 = N^2$$

where

N=Ni.N2 is the total number of data points.

Thus, in general, the total number of operations in the direct calculation of Fourier transform equals the square of the total number of data points under consideration.

For $N=2^{12}$, for example, the tatal number of computations needed equals 2^{24} $1.7xio^7$. Since each calculation requires about 20 sec. on IBM 7044, the computation time required is about 340 secs.

Presented in the following section is a method based on Cooley & Tukey algorithm which requires $N\log_2 N$ computations, thereby, reducing the computation time considerably. For example, for $N=2^{12}$, the time required is on only one sec. as against 340 secs. required by direct computation

6. COOLEY TUKEY ALGORITHM

6.i One Dimensional Case:

The Fourier transform in the one dimensional case can be represented as:

$$X(J)=\sum_{K=0}^{N-1} A(K)W^{JK} \qquad \text{for } J=o,i,2,\ldots\ldots,N-i$$

where

A'^{s} are the given complex numbers.& $W=\exp(2\pi j/N)$

If $N=2^{M}$=Number of data points

J can be written in the binary form as:

$$J=J_{M-i}2^{M-i}+\ldots\ldots\ldots\ldots+J_o$$

where $J_o,J_i,\ldots\ldots\ldots,J_{M-i}$ is either 0 or I.

Writing in the same way,

$$K=K_{M-i}2^{M-i}+\ldots\ldots\ldots\ldots+K_o$$

The Fourier transform can be represented in the following form:

$$X(J_{M-i},\ldots\ldots,J_c)=\sum_{K_0=0}^{1}\cdots\sum_{K_{M-1}=0}^{1} A(K_{M-i},\ldots,K_o).\; W^{JK_{M-i}2^{M-i}}\ldots W^{JK_o} \qquad (6)$$

Since, $W^{2^{M}}=\exp(2\pi j)=i$

$$W^{JK_{M-i}2^{M-i}}=W^{(J_{M-i}2^{M-i}+\ldots\ldots+J_o)K_{M-i}2^{M-i}}=W^{J_oK_{M-i}2^{M-1}}$$

The innermost sum in (6) depends entirely on J_o, $K_{M-2},\ldots\ldots,K_o$.Letting $A_i(J_o,K_{M-2},\ldots\ldots,K_o)$ denote this sum, and defining $A_2(J_o,J_i,K_{M-3},\ldots\ldots,K_o)$ as the next innermost sum over K_{M-2} and so on,we obtain the algorithm:

$$A_i(J_o,K_{M-2},\ldots,K_o)=\sum_{K_{M-1}=0}^{1} A(K_{M-i},\ldots\ldots,K_o)W^{J_oK_{M-i}2^{M-i}}$$

$$A_2(J_o,J_i,K_{M-3},\ldots,K_o)=\sum_{K_{M-2}=0}^{1} A_i(J_o,K_{M-2},\ldots\ldots,K_o).\; W^{(J_i.2+J_o)K_{M-2}.2^{M-2}}$$

$$A_L(J_o,J_i,\ldots,J_{L-i},K_{M-L-i},\ \ldots,K_o)$$

$$= \sum_{K_{M-L}=0}^{1} A_{L-i}(J_o,J_i,\ldots,J_{L-2},K_{M-L},K_{M-L-i},\ldots\ldots K_o). \qquad (7)$$
$$W^{(J_{L-i}2^{L-i}+\ldots\ldots+J_o)K_{M-L}2^{M-L}}$$

since,
$$W^{2^{M-i}}=-i$$

$$A_L(J_o,J_i,\ldots,J_{L-i},K_{M-L-i},\ldots,K_o)$$
$$=A_{L-i}(J_o,\ldots\ldots,J_{L-2},\underline{0},K_{M-L-i},\ldots\ldots,K_o)+$$

$$(-i)^{J_{L-i}}A_{L-i}(J_o,\ldots,J_{L-2},\underline{i},K_{M-L-i},\ldots\ldots,K_o).$$
$$W^{(J_{L-2}2^{L-2}+\ldots\ldots+J_o)2^{M-L}}$$

for $J_{L-i}=o,i$ and all values of $J_o,\ldots,$
J_{L-2}. and $K_{M-L-i},\ldots\ldots,K_o$.

Calculating for $L=i,2,\ldots\ldots\ldots,M$ yields

$$X(J_{M-i},\ldots\ldots,J_o)=A_M(J_o,\ldots\ldots,J_{M-i}) \qquad (8)$$

The implication of (8) is that the bit configuration of the index of A_M is to be put in the reversed order in order to obtain the index of the corresponding X

At successive stages the exponent of W represented in binary are:

$$K_{M-i}(J_o,o,\ldots\ldots,o)=(J_o2^{M-i}+o+\ldots\ldots+o).K_{M-i}$$
$$K_{M-2}(J_i,J_o,o,\ldots,o)=(J_i2^{M-i}+J_o2^{M-2})K_{M-2}$$
.

.

..

$$K_{M-L}(J_{L-i},\ldots,J_o,o\ldots,o)=(J_{L-i}2^{M-i}+\ldots+J_o2^{M-L})K_{M-L}$$

Thus W^{JK} has been factored as:
$$W^{JK}=W^{JK \bmod N}$$

$$= W^{K_{M-1}(J_0,0,\dots,0)} \cdot W^{K_{M-2}(J_1,J_0,0\dots,0)} \cdots$$

$$W^{K_{M-L}(J_{L-1},\dots,J_0,0..0)} \cdots W^{K_0(J_{M-1},\dots,J_0)}$$

The tree-graph representation corresponds to the factoring of W^{JK} as explained above.

Example

Let us consider a one dimensional example using 8 data points

since $$A_1(J_0,K_{M-2},\dots,K_0) = \sum_{K_{M-1}=0}^{1} A(K_{M-1},\dots\dots,K_0) \cdot W^{(J_0 . K_{M-1} . 2)}$$

For N=8

M=3

$$A_1(J_0,K_1,K_0)=A(\underline{0},K_1,K_0)+(-1)^{J_0}A(\underline{1},K_1,K_0)$$

for $J_0=0,1$ and all other values of k_1,K_0.

Substituting all possible values of J_0,k_1,K_0, we obtain:

$$A_1(000)=A(000)+A(100)$$
$$A_1(100)=A(000)-A(100)$$

$$A_1(010)=A(010)+A(110)$$
$$A_1(110)=A(010)-A(110)$$

$$A_1(001)=A(001)+A(101)$$
$$A_1(101)=A(001)-A(101)$$

$A_1(011)=A(011)+A(111)$

$A_1(111)=A(011)-A(111)$

<u>2nd Summation</u>:

$$A_2(J_0,J_1,K_0)=\sum_{K_1=0}^{1} A_1(J_0,K_1,K_0).W^{(J_1.2+J_0)K_1.2}$$

Calculating for $J_1=0,1$ & all other values of J_0,K_0, we obtain:

$A_2(000)=A_1(000)+A_1(010)$

$A_2(010)=A_1(000)-A_1(010)$

$A_2(001)=A_1(001)+A_1(011)$

$A_2(011)=A_1(001)-A_1(011)$

$A_2(100)=A_1(100)+A_1(110).W_2^2$

$A_2(110)=A_1(100)-A_1(110).W_2^2$

$A_2(101)=A_1(101)+A_1(111).W_2^2$

$A_2(111)=A_1(101)-A_1(111).W_2$

Once A_L's are computed, A_{L-1}'s are no more needed. The same space, which was previously occupied by A_{L-1}'s, can now be alloted to A_L's. Moreover, since each A_{L-1} appears only twice in the calculation of the corresponding A_L's, these two A_L's can be computed first and can now occupy the place of A_{L-1}'s which have been exhausted thus requiring only two more locations for the complete storage of $A_L(K)$.

<u>3rd Summation</u>:

$$A_3(J_0,J_1,J_2)=\sum_{K_0=0}^{1} A_2(J_0,J_1,K_0)W^{(J_2 2^2+J_1 2+J_0)K_0}$$

Substituting the values of $A_2(K)$, we obtain:

$$A_3(0)=A(0)+A(1)+A(2)+A(3)+A(4)+A(5)+A(6)+A(7)$$

$$A_3(1)=A(0)-A(1)+A(2)-A(3)+A(4)-A(5)+A(6)-A(7)$$

$$A_3(2)=A(0)+A(1)W^2-A(2)-A(3)W^2+A(4)+A(5)W^2 \\ -A(6)-A(7)W^2$$

$$A_3(3)=A(0)-A(1)W^2-A(2)+A(3)W^2+A(4)-A(5)W^2 \\ -A(6)+A(7)W^2$$

$$A_3(4)=A(0)+A(1)W+A(2)W^2+A(3)W^3-A(4)-A(5)W \\ -A(6)W^2-A(7)W^3$$

$$A_3(5)=A(0)-A(1)W+A(2)W^2-A(3)W^3-A(4)+A(5)W \\ -A(6)W^2+A(7)W^3$$

$$A_3(6)=A(0)+A(1)W^3-A(2)W^2-A(3)W-A(4)-A(5)W^3 \\ +A(6)W^2-A(7)W$$

$$A_3(7)=A(0)-A(1)W^3-A(2)W^2-A(3)W-A(4)+A(5)W^3 \\ +A(6)W^2+A(7)W$$

In the matrix notation, the various summations can be represented as follows:

$$\begin{bmatrix} A_1(000) \\ A_1(001) \\ A_1(010) \\ A_1(011) \\ A_1(100) \\ A_1(101) \\ A_1(110) \\ A_1(111) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & W^0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W^0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W^0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W^0 \\ 1 & 0 & 0 & 0 & W^4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W^4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W^4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W^4 \end{bmatrix} \begin{bmatrix} A(000) \\ A(001) \\ A(010) \\ A(011) \\ A(100) \\ A(101) \\ A(110) \\ A(111) \end{bmatrix}$$

$$\begin{bmatrix} A_2(000) \\ A_2(001) \\ A_2(010) \\ A_2(011) \\ A_2(100) \\ A_2(101) \\ A_2(110) \\ A_2(111) \end{bmatrix} = \begin{bmatrix} 1 & 0 & W^0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & W^0 & 0 & 0 & 0 & 0 \\ 1 & 0 & W^4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & W^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & W^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & W^2 \\ 0 & 0 & 0 & 0 & 1 & 0 & W^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & W^6 \end{bmatrix} \begin{bmatrix} A_1(000) \\ A_1(001) \\ A_1(010) \\ A_1(011) \\ A_1(100) \\ A_1(101) \\ A_1(110) \\ A_1(111) \end{bmatrix}$$

$$\begin{bmatrix} X(000) \\ X(100) \\ X(010) \\ X(110) \\ X(001) \\ X(101) \\ X(011) \\ X(111) \end{bmatrix} = \begin{bmatrix} A_3(000) \\ A_3(001) \\ A_3(010) \\ A_3(011) \\ A_3(100) \\ A_3(101) \\ A_3(110) \\ A_3(111) \end{bmatrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & W^4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & W^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & W^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & W^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & W^5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & W^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & W^7 \end{bmatrix} \begin{bmatrix} A_2(000) \\ A_2(001) \\ A_2(010) \\ A_2(011) \\ A_2(100) \\ A_2(101) \\ A_2(110) \\ A_2(111) \end{bmatrix}$$

The vector $\left[X(n)\right]$ is merely $\left[A_3(n)\right]$ with the binary argument flipped, therefore, by bit reversing the argument of vector $\left[A_3(n)\right]$ one knows exectly where each component of the vector $\left[X(n)\right]$ actually belongs. The general form of the factored matrices has been developed by E.O.Brigham & R.E.Morrow^2. This is discussed in Appendix-E.

6.2 <u>Two Dimensional Case</u>:

Letting

~~N~~

$N_a=2^{M_a}$ be the number of points in the $a=i,2$ directions. Defining:

$Mi'=Mi$

$M2'=Mi+M2$

we have the indices Ji, J2 in the bit positions 0 to Mi'-i, Mi' to M2'-i, respectively, of an index J. Thus the bit configuration of J is described by the bit positions:

$$M2',\underbrace{M2'-i,\ldots\ldots,Mi'}_{J2},\underbrace{Mi'-i,\ldots\ldots,0}_{J1}$$

$$J=J_{M2'-i}\cdot 2^{M2'-i}+\ldots+J_{Mi'}\cdot 2^{Mi'}+J_{Mi'-i}\cdot 2^{Mi-i}+\ldots+J_0$$

The bit configuration for K may similarly be written as:

$$K=K_{M2'-i}2^{M2'-i}+\ldots+K_{Mi'}2^{Mi'}+K_{Mi'-i}2^{Mi'-i}+\ldots+K_0$$

The two dimensional Fourier transform may thus be represented as follows:

$$X(J_2,J_i)=\sum_{K'_{M2-1}=0}^{1}\cdots\sum_{K'_{Mi}=0}^{1}\sum_{K'_{Mi-1}=0}^{1}\cdots\sum_{K_0=0}^{1} A(K_2,K_i)\,.\,Wi^{(J_{Mi'-i}2^{Mi'-i}+\ldots+J_0)K_i}\,W2^{(J_{M2'-i}2^{M2'-i}+\ldots+J_{Mi'}2^{Mi'})K_2} \qquad (10)$$

where

$$W_a=\exp(2\pi j/N_a),\ a=I,2$$

The Lth summation over KI may be written from (7) in the following form:

$$A_L(K_{M2'-i},\ldots,K_{Mi'},J_o,\ldots,J_{L-2};\overset{\downarrow}{\underline{J_{L-i}}},K_{Mi'-L-i},\ldots,K_o)$$

$$= \sum_{K_{M-L}=0}^{i} A_{L-i}(K_{M2'-i},\ldots,K_{Mi'},J_o,\ldots,J_{L-2},\overset{\downarrow}{\underline{K_{Mi'-L}}},\ldots,K_o).\; W_i^{(J_{L-i}2^{L-i}+\ldots+J_o)K_{Mi'-L}2^{Mi'-L}} \qquad (11)$$

Letting $\bar{J}=p.2^{Mi}+q.2^{Mi-L+i}+r$

where

$p=0,I,2,\ldots\ldots,(2^{M2-i}-i)$

$q=0,I,2,\ldots\ldots,(2^{L-i}-i)$

$r=0,I,2,\ldots\ldots,(2^{Mi-L}-i)$

Putting in the values for $p,q,r,\bar{J}$, we obtain:

$$A_L(\bar{J}+J_{L-i}2^{Mi-L})=A_{L-i}(\bar{J})+A_{L-i}(\bar{J}+2^{Mi-L})\; W_i^{(\bar{Q}+J_{L-i}2^{L-i})2^{Mi-L}} \qquad (12)$$

where

$Q=J_o2^{L-2}+\ldots\ldots+J_{L-2}$

$\bar{Q}=J_{L-2}2^{L-2}+\ldots\ldots+J_o$

Solving for $J_{L-i}=o,i$, we obtain

$$A_L(\bar{J})=A_{L-i}(\bar{J})+A_{L-i}(\bar{J}+2^{Mi-L})\,W_i^{\bar{Q}.2^{Mi-L}} \qquad (13)$$

$$A_L(\bar{J}+2^{Mi-L})=A_{L-i}(\bar{J})-A_{L-i}(\bar{J}+2^{Mi-L})\,W_i^{\bar{Q}.2^{Mi-L}} \qquad (14)$$

since

$$W_i^{(2^{Mi}-i)}=-i$$

__2nd Summation:__

Letting A'denote the result of the first summation

$$A'=A_{Mi}(K'_{M2-i},\ldots,K_{Mi},J_o,\ldots,J_{Mi-i})$$

$$X(J2,Ji)=\sum_{K2=0}^{M2-1} A'.W2^{J2K2}$$

$$=\sum_{K_{M2'-1}=0}^{1}\cdots\sum_{K_{Mi'}=0}^{1} A'(K_{M2'-i},\ldots,K_{Mi},J_o,\ldots,J_{Mi-i}).\ W2^{J2K2} \qquad (15)$$

Performing the summation over $K_{M2'-L}$ and denoting it by A'_L, we obtain:

$$A'_L(J_{Mi},\ldots,\underline{J_{Mi+L-i}},K_{M2'-L-i},\ldots,K_{Mi},J_o,\ldots,J_{Mi-i})$$

$$=\sum_{K_{M2'-L}=0}^{1} A_{L-i}(J_{Mi},\ldots,J_{Mi+L-2},\underline{K_{M2'-L}},\ldots,K_{Mi},J_o,\ldots,J_{Mi-i}).\ W2^{(J_{Mi+L-i}2^{L-i}+\ldots+J_{Mi})K_{M2'-L}2^{M2-L}} \qquad (16)$$

Letting

$$\bar{J}=p.2^{M2'}+q.2^{M2'-L-i}+r$$

where

$$p=o$$

$$q=J_{Mi}2^{L-2}+J_{Mi+i}2^{L-3}+\ldots\ldots+J_{Mi+L-2}$$

$$\bar{q}=J_{Mi+L-2}2^{L-2}+J_{Mi+L-3}2^{L-3}+\ldots\ldots+J_{Mi}$$

$$r=(K_{M2'-L+i}2^{M2-L+i}+\ldots\ldots+K_{Mi})2^{Mi}+(J_o2^{Mi-i}+\ldots+J_{Mi-i}$$

Substituting for J_{Mi+L-i}=o,i,we obtain the algorithm:

$$A'_L(\bar{J})=A'_{L-i}(\bar{J})+A'_{L-i}(\bar{J}+2^{M2'-L})W2^{\bar{q}.2^{M2-L}} \qquad (17)$$

$$A'_L(\bar{J}+2^{M2'-L})=A'_{L-i}(\bar{J})-A'_{L-i}(\bar{J}+2^{M2'-L})W2^{\bar{q}.2^{M2-L}} \qquad (18)$$

where

$p=0,I,2,.....,(2^{M2'-Ma'}-i)$,

corresponding to bit positions: $M2'-i,...,Ma'$

$q=0,I,2,.....,(2^{L-i}-i)$,

corresponding to the bit positions:

$Ma'-i,.....,Ma'-L+i$

$r=0,I,2,.....,(2^{Ma'-L}-i)$,

corresponding to the bit positions:

$Ma'-L-i,.....,I,0$

$a=I,2$,

corresponding to the two dimensions respectively

6.3 Computational Time Involved In Cooley & Tukey Algorithm:

Time involved in Cooley & Tukey algorithm is best explained by reffering to the generalized factored matrices representation of the algorithm. As there are only two non zero entries in each row, and one of these two entries is unity, therefore, in each matrix multiplication the number of complex multiplications and additions needed are respectively, N/2 & N, since in half of the terms, the complex multiplication needed is already carried out in the other half of the terms ($W^{n+N/2}=-W^n, n=N/2$). Thus the total number of complex multiplications and additions needed are, respectively, $\frac{1}{2}N.M$ & $N.M$ ($N=2^M$). The computation time needed in Cooley & Tukey algorithm is thus roughly proportional to $N\log_2 N$, where N is the total number of data points.
For $N=2^{I2}$, the time required on IBM 7044 works out to be nearly one second.

Thus through the use of this algorithm the time requirement has been considerably reduced for large arrays (one second as against 340 seconds for direct computation on IBM 7044 for 4096 samples).

7. FORTRAN IV PROGRAM FOR IBM 7044

7.I Options:

The program for Cooley & Tukey algorithm has been written in Fortran language using the least sophistication level. The program can be made much more efficient by the use of MAP subroutines which offer the following distinct advantages over Fortran IV:

(a) Compilation time is saved since Fortran IV is first converted into MAP and then processed.

(b) A lot of time can be saved by the use of logical operations such as bit inversion, multiplication by 2^n, binary addition. These operations are allowed only in MAP.

(c) Computer storage can be more effectively utilized

The Fortran program allows for calculating the Fourier transform as well as the inverse transform of one dimensional as well as two dimensional data (see Appendix-F for control cards). Print statements may be added to print out INTERMEDIATE location indices to identify the flow of the program at each summation stage.

7.2 Capacity:

Since IBM 7044 has 32K memory and since the program handles data points which are a power of 2 ($2^{I2}, 2^{I3}, \ldots\ldots,$), allowing for arrays AR(I), AI(I), FR(ISC), FI(ISC) a total storage of I6K, which leaves I6K for program instructions, the maximum array size that can be handled is thus 4096 data points. The computer may more efficiently be used by one of the following techniques:

(a) <u>Simultaneous Fourier analysis of two sets of real data:</u>

$$X(J)=X'(J)+jX''(J)=\sum_{K=0}^{N-1} A(K)W^{JK} \quad \text{for } J=0,I,..,N-I \tag{19}$$

where

$X'(J)$ & $X''(J)$ are real.

The complex amplitudes of

$$X'(J)=\sum_{K=0}^{N-1} A'(K)W^{JK}$$

$$X''(J)=\sum_{K=0}^{N-1} A''(K)W^{JK}$$

are given by (Appendix-C)

$$A'(K)=\tfrac{1}{2}\left[A(K)+A^{*}(N-K)\right] \qquad (20)$$

$$A''(K)=\tfrac{1}{2}\left[A(K)-A^{*}(N-K)\right] \qquad (21)$$

Therefore, given two sets of real data, one can let these be the real and imaginary parts of the complex data and then the inversion formula can be used to obtain complex A'^{s}. These, used in a two dimensional of (19), will give the desired amplitude of (20) & (21).

(b) <u>To Double the number of data points</u>:

A Fourier analysis of a large number of data points (larger than 2^{12}) can be performed by the use of external storage and multiple references to the program. The procedure for doubling the number of data points is as follows:

Let N be the number of points and N/2 be the capacity of the core memory available. Let

$$W=\exp(2\pi j/N) \quad \text{so that} \quad W^2=\exp(2\pi j/\tfrac{1}{2}N)$$

Then, in two separate passes with the program, one can calculate the Fourier coefficients of the two N/2 Fourier transforms:

$$X(2J)=\sum_{K=0}^{N/2-1} A'(K)(W^2)^{JK}$$

$$\text{for } J=0,1,2,\ldots,\tfrac{1}{2}N-1$$

$$X(2J+1)\ \sum_{K=0}^{N/2-1} A''(K)(W^2)^{JK}$$

The coefficients of

$$X(J)=\sum_{K=0}^{N-1} A(K)W^{JK} \qquad \text{for } J=0,1,2,\ldots,N-1$$

are given by (Appendix-D)

$$A(K)=\tfrac{1}{2}\left[A'(K)+A''(K)W^{-K}\right]$$

$$\text{for } K=0,1,2,\ldots,\tfrac{1}{2}N-1$$

$$A(\tfrac{1}{2}N+K)=\tfrac{1}{2}\left[A'(K)-A''(K)W^{-K}\right]$$

Repetition of the above procedure permits successive doubling of the capacity of the program.

7.3 Truncation Error:

To test the accuracy of the program, several input waveforms were transformed. The inverse transformation was performed on the result and the r.m.s difference between the final and initial data was computed. The results obtained were as follows:

N	Type of data	R.M.S Error
64	square wave	10^{-8}
4096	sinx/x	10^{-6}

8. RESULTS OF HOLOGRAM GENERATION

Two dimensional Fourier Transform Hologram was generated corresponding to a completely white transparency (each sample has unity amplitude).The Fourier transform of such a picture corresponds to an impulse at $\omega = 0$, as is clear from the Fourier transform results. The reference signal consisted of a plane wave a minimum ten times brighter than the picture's brightest sample in order to keep the reconstruction error with in 1% .This error is inversely proportional to the square of the brightness of the reference beam relative to the picture's brightest signal samlpe.The relative figure of ten was assumed in this particular example.The hologram thus generated was inverse Fourier transformed to get the original signal.An error of the order of 10^{-6} was obtained for 8 points.

```
        DATA FOR TWO DIMENSIONAL HOLOGRAM SYNTHESIS
1.00000000    1.00000000    1.00000000    1.00000000
1.00000000    1.00000000    1.000000000   1.00000000
1.00000000    1.00000000    1.00000000    1.00000000
1.00000000    1.00000000    1.00000000    1.00000000
1.00000000    1.00000000    1.00000000    1.00000000
1.00000000    1.00000000    1.00000000    1.00000000
1.00000000    1.00000000    1.00000000    1.00000000
1.00000000    1.00000000    1.00000000    1.00000000
                     FOURIER TRANSFORM
   REAL        IMAGINARY     ABSOLUTE        REAL       IMAGINARY     ABSOLUTE
32.0000000    0.00000000    32.0000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
0.00000000    0.00000000    0.00000000    0.00000000    0.0000000    0.0000000
```

```
                          GENERATED HOLOGRAM

1764.00000000   99.99999905   99.99999714   99.99999714
  99.99999714   99.99999714   99.99999809   99.99999809
 100.00000000   99.99999619   99.99999714   99.99999619
  99.99999619   99.99999714   99.99999614   99.99999714
 100.00000000   99.99999619   99.99999714   99.99999619
  99.99999714   99.99999619   99.99999714   99.99999714
 100.00000000   99.99999619   99.99999619   99.99999619
  99.99999809   99.99999619   99.99999619   99.99999905

                           RECONSTRUCTION

     REAL          IMAGINARY         REAL        IMAGINARY
151.99999619    0.00000000     51.99999857   -0.00000011
 51.99999952   -0.00000008     51.99999809    0.00000001
 51.99999952    0.00000008     51.99999905   -0.00000004
 51.99999952   -0.00000008     51.99999905   -0.00000009
 52.00000000    0.00000000     51.99999857    0.00000011
 51.99999857    0.00000008     51.99999762    0.00000000
 51.99999905    0.00000008     51.99999857   -0.00000012
 51.99999857    0.00000008     51.99999906   -0.00000008
 52.00000000    0.00000000     51.99999857    0.00000008
 51.99999857   -0.00000008     51.99999809    0.00000012
 51.99999952   -0.00000008     51.99999809    0.00000000
 51.99999857   -0.00000008     51.99999905   -0.00000011
 52.00000000    0.00000000     51.99999857    0.00000009
 51.99999952    0.00000008     51.99999857    0.00000004
 52.00000000   -0.00000008     51.99999857   -0.00000001
 51.99999952    0.00000008     51.99999905    0.00000011
```

9. OTHER APPLICATIONS OF FFT PROGRAM PACKAGE

The FFT program has also been employed to find the convolution integral and auto-correlation function[3]. Another important application of FFT program is in the synthesis of antenna patterns with equally spaced elements. This particular application of FFT is discussed below:

For a general linear array with equally spaced elements, the relative amplitude of the radiated field intensity is given by:

$$E=I_0+I_1e^{j\psi}+I\ldots\ldots\ldots\ldots\ldots+I_{n-1}e^{j(n-1)\psi} \qquad (22)$$

where

I_K are the complex element currents,

$\psi=\frac{2\pi}{\lambda}.d\cos\phi+\alpha$

d=spacing between successive elements

and α=progressive phase shift from left to right

Eq.(22) can be written as:

$$E=\sum_{K=0}^{n-1} I_K Z^K$$

where

$Z=e^{j\psi}$

If we take discrete samples of E(J) at regular intervals, then the phase associated with the Jth sample is:

$\frac{2\pi}{n}.J,\ J=0,1,2,\ldots\ldots,n-1$

The intensity for the Jth sample may be written in the following form:

$$E(J)=\sum_{K=0}^{n-1} I_K W^{JK} \qquad (23)$$

where

$W=e^{j2\pi/n}$

Eq.(23) is the complex Fourier inverse of the function I_K. The required complex current amplitudes can be found by the Fourier transform of the discrete function E(J)

The complex current amplitudes are thus given by:

$$I_K = \frac{1}{n} \sum_{J=0}^{n-1} E(J) W^{-JK} \qquad (24)$$

Thus, a given antenna pattern can be transformed into the domain, discretized and Fourier transformed to get the complex current amplitudes, I_K, which produce the desired radiation pattern. The specific case of an end fire array with $\lambda/2$ spacing between successive elements is discussed below:

For $d = \lambda/2$

$\psi = \pi \cos\phi + \alpha$

For an end fire array:

$\alpha = -\pi$

$\psi = \pi(\cos\phi - 1)$

The simple pattern shown in Fig.9·1 when transformed into the domain looks like Fig.9·2

$$f(\psi) = \left[\frac{1}{2} + \sum_{n=0}^{\infty} \frac{\sin(n\pi/2)}{n\pi/2} . \cos(n\psi)\right] \qquad (25)$$

which is the Fourier series expansion of the function $f(\psi)$. This pattern has been synthesized for N=64 using $\frac{\sin K\pi/2}{K\pi/2}$, K=0,1,2,.....,63, as the input data.

Similarly, synthesis has been carried out for 64 element two dimensional array using $\frac{\sin K'\pi/2}{K'\pi/2} \cdot \frac{\sin K''\pi/2}{K''\pi/2}$, K'=0,1,2,...,7, K"=0,1,2,...,7, as the two dimensional data. The results thus obtained have been plotted against the ideal patterns expected for $N=\infty$.

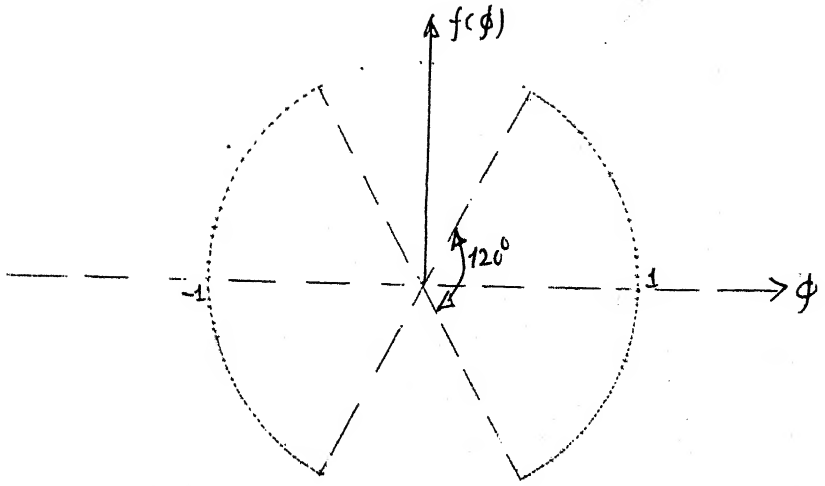


Fig 9.1

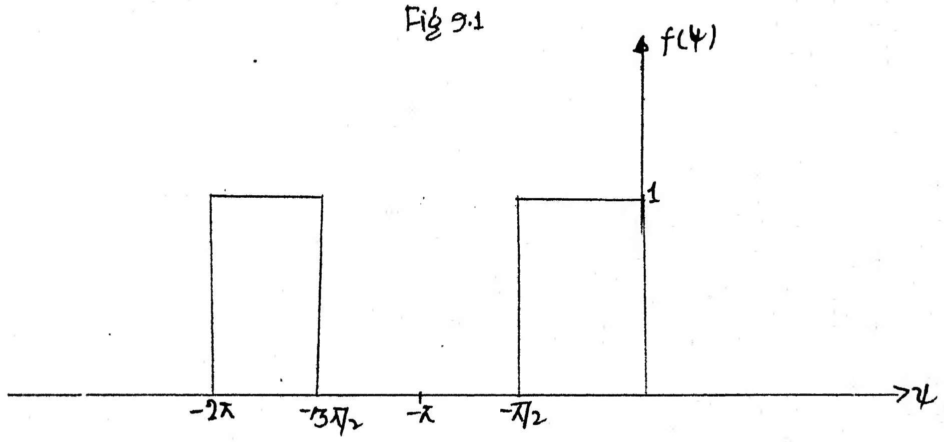


Fig 9.2.

10. CONCLUSIONS

One and two dimensional Fourier Transform Hologram were successfully generated & reconstructed on IBM 7044 using FFT program package. The results are presented in sec.8. Synthesis technique for one & two dimensional antenna patterns for equally spaced elements using FFT program package has also been successfully demonstrated.

The value of the FFT program lies in the reduction of the computer time. An N-point transformation by the direct method requires a time proportional to N^2 where as the FFT program requires a time proportional to $N.\log_2 N$. The approximate time saving factor is thus given by:

$$N/\log_2 N$$

Other importent applications of the FFT program package consists in finding the convolution integral, the auto-correlation function etc. It has made possible analysis in the frequency domain which was considered impracticable otherwise. This may find enormous applications in band width reduction schemes, character recognition, spatial filtering.

REFERENCES:

1.Cooley,J.W.,and Tukey,J.W.,"an algorithm for the machine calculation of complex Fourier series,"Math.Comput.,vol.19, pp.297-301,April1965.

2.Good,I.J.,"The interaction algorithm for practical Fourier analysis" J.Roy.Statist.Soc.,ser.B,vol 20,pp. 361-372,1958

3.Stockham,T.G.,"High-speed convolution and correlation," 1966 Joint Computer Conference,AFIPS Proc.,vol.28,pp.229-233.

4.Brighm,E.O.,Morrow,R.E.,"The fast Fourier transform," IEEE Spectrum,pp.63-70,Dec.1967

5.Moharir,P.S.,"convolution integration using FFT technique," Term paper,EE 661.

6 Huang,T.S.,Prasada,B.,"Considerations on the generation & processing of Holograms by digital computers," M.I.T Research Lab. of Electronics, QPR No. 81,pp.199-205

APPENDIX-A

Programming Details

The Lth summation over the first dimension is given by:

$$A_L(\underleftarrow{K_{M2'-i},.....,K_{Mi}},\underrightarrow{J_o,J_i,....,J_{L-i}},\underleftarrow{K_{Mi-L-i},......,K_o})$$

$$= A_{L-i}(K_{M2-i},....,K_{Mi},J_o,.....,J_{L-2},0,K_{Mi-L-i},...,K_o)+$$

$$(\quad (-i)^{J_{L-i}} A_{L-i}(K_{M2-i},...,K_{Mi},J_o,...J_{L-2},i,K_{Mi-L-i},..,K_o).$$

$$W_i^{(J_{L-2}.2^{L-2}+.......+J_o).2^{Mi-L}}$$

The Lth summation over the 2nd dimension is given by:

$$A_L(\underrightarrow{J_{Mi},.....,J_{Mi+L-2}},\overset{\downarrow}{J_{Mi+L-i}},\underleftarrow{K_{M2'-L-i},...,K_{Mi}},\underrightarrow{J_o,J_i,..,J_{Mi-i}})$$

$$= A_{L-i}(J_{Mi},....,J_{Mi+L-2},0,K_{M2-L-i},...,K_{Mi},J_o,J_i,...,J_{Mi-i})$$

$$+(-i)^{J_{Mi+L-i}} A_{L-i}(J_{Mi},..,J_{Mi+L-i},i,K_{M2-L-i},..,K_{Mi},J_o,.., J_{Mi-i}).$$

$$W_2^{(J_{Mi+L-i}.2^{L-i}+......+J_{Mi}).2^{M2-L}}$$

The final summation over the two dimensions gives the Fourier transform with inverted bit configuration, i.e.,

$$X(\underleftarrow{J_{M2'-i},........,J_{Mi}},\underleftarrow{J_{Mi-i},........,J_o})$$

$$=A_{M2}(\underrightarrow{J_{Mi},.....,J_{M2'-i}},\underrightarrow{J_o,.....,J_{Mi-i}})$$

From the last equation, it is clear that if the final transform has to be in the proper bit configuration, the summation at each stage has to be carried out with the required number of bits in the inverted configuration. This is shown by arrows in the above algorithm. The arrows point towards the bits requiring the least number of changes at each stage.This is illustrated below in the case of a 8x8 picture plane requiring three bits in each direction.

$$
\begin{array}{ll}
A_i(\underline{K_5,K_4,K_3},\overset{\uparrow}{J_o},K_i,K_o)=(&)+A(K_5,K_4,K_3,0,K_i,K_o)+ \\
\longleftarrow \qquad \longleftarrow \qquad (&).A(K_5,K_4,K_3,i,K_i,K_o) \\
A_2(\underline{\underline{K_5,K_4,K_3}},J_o,\overset{\downarrow}{J_i},K_o)=(&).A_i(K_5,K_4,K_3,J_o,0,K_o)+ \\
\Longleftarrow \qquad \qquad (&).A_i(K_5,K_4,K_3,J_o,i,K_o) \\
A_3(K_5,K_4,K_3,J_o,J_i,\overset{\downarrow}{J_2})=(&).A_2(K_5,K_4,K_3,J_o,J_i,0)+ \\
\Longleftarrow \qquad \longrightarrow \qquad (&).A_2(K_5,K_4,K_3,J_o,J_i,i) \\
 & \\
A_i(\overset{\downarrow}{J_3},\underline{K_4,K_3},J_o,J_i,J_2)=(&).A_3(0,K_4,K_3,J_o,J_i,J_2)+ \\
\Longleftarrow \qquad \longrightarrow \qquad (&).A_3(i,K_4,K_3,J_o,J_i,J_2) \\
A_2(J_3,J_4,K_3,J_o,J_i,J_2)=(&).A_i(J_3,0,K_3,J_o,J_i,J_2)+ \\
\Longrightarrow \qquad \longrightarrow \qquad (&).A_i(J_3,i,K_3,J_o,J_i,J_2) \\
A_3(J_3,J_4,\overset{\downarrow}{J_5},J_o,J_i,J_2)=(&).A_2(J_3,J_4,0,J_o,J_i,J_2)+ \\
\Longrightarrow \qquad \longrightarrow \qquad (&).A_2(J_3,J_4,i,J_o,J_i,J_2)
\end{array}
$$

The arrow on the top indicates the bit over which the summation is being performed.The problem is then reduced to calculating proper indices at each summation.

Let PP(I) denote the bit positions in the 2nd dimension over which the summation is yet to be carried out,QQ(I)denote the bit positions in the 2nd dimension over which the summation has already been performed. Similarly,let R(I) denote the bit positions in the ist dimension over which the summation is yet to be carried out, Q(I) denote the bit positions in the ist dimension over which the summation has been carried out.The indices of the inverted bit configuration at any stage can then be represented as:

$$
J=\left[\sum_{I=1}^{J-1} QQ(I).2^{M2-I}+\sum_{I=1}^{M2-J} PP(I).2^{I-i}\right]Ni+\left[\sum_{I=1}^{L-1} Q(I).2^{Mi-I}+\sum_{I=1}^{Mi-L} R(I).2^{I-i}\right]
$$

and

$J+2^{Mi-L}$ or $J+\left[2^{M2-J}\right].Ni$

where L<u>th</u> summation in the first dimension and the J<u>th</u> summation in the second dimension are being considered.

APPENDIX-B

The one dimensional Fourier transform pair is defined as:

$$A(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}X(j\omega)e^{j\omega t}d\omega$$

$$X(j\omega)=\int_{-\infty}^{\infty}A(t)e^{-j\omega t}dt$$

If $A(t)$ is periodic with petiodicity, T, the transform pair becomes:

$$A(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}X(j\omega)e^{j\omega t}\delta(\omega-n\omega_0)d\omega=\sum_{n=-\infty}^{\infty}X(jn\omega_0)e^{jn\omega_0 t}$$

where

$$\omega_0=2\pi/T$$

If $A(t)$ is sampled at intervals Δt

$$t=K\Delta t$$

$$A(K\Delta t)=\quad X(jn\omega_0)e^{jn2\pi K\Delta t/T}$$

Letting

$$N=T/\Delta t$$

$$A(K\Delta t)=\sum_{n=-\infty}^{\infty}X(jn\omega_0)e^{jn2\pi K/N}$$

$$=\sum_{n=-\infty}^{\infty}X(jn\omega_0)W^{nK}$$

where

$$W=e^{j2\pi/N}$$

$A(K\Delta t)$ representsthe amplitude of the Kth sample and $X(jn\omega_0)$ the complex amplitude of the nth Fourier coefficient. These will henceforth be abbreviated as: $A(K)$ & $X(n)$

$$A(K)=\sum_{n=-\infty}^{\infty}X(n)W^{nK}$$

For the periodic, sampled signals

$$X(n)=\frac{1}{T}\int_{0}^{T}A(t)e^{-jn\omega_0 t}dt$$

$$=\frac{1}{T}\int_{0}^{T}A(K\Delta t)e^{-jn\omega_0 K\Delta t}\,\delta(t-K\Delta t)dt$$

$$=\frac{1}{T}\sum_{K=0}^{N-1}A(K)e^{-jn2\pi K\Delta t/T}$$

$$=\frac{1}{N\Delta t}\sum_{K=0}^{N-1}A(K)W^{-nK}$$

According to the sampling theorem, any function $A(t)$ band limited to B cycles is completely described by the sample values i/2B sec. apart.

$$t = i/2B$$
$$= i/2n_{max} \cdot f_o = T/2n_{max}$$

or
$$n_{max} = T/2\Delta t = \tfrac{1}{2}N$$

Thus frequency band from $-\tfrac{1}{2}N+i$ to $\tfrac{1}{2}N$ is sufficient to completely describe the signal.

$$A(K) = \sum_{-n_{max}+1}^{n_{max}} X(n)W^{nK} = \sum_{n=-\frac{1}{2}N+1}^{N/2} X(n)W^{nK}$$
$$= \sum_{n=-N/2+1}^{1} X(n)W^{nK} + \sum_{n=0}^{N/2} X(n)W^{nK}$$

$$\sum_{n=-N/2+1}^{1} X(n)W^{nK} = \sum_{m=N/2+1}^{N-1} X(m-N)W^{(m-N)K} = \sum_{n=N/2+1}^{N-1} X(n-N)W^{nK}$$
$$= \sum_{n=N/2+1}^{N-1} \frac{W^{nK}}{N\Delta t} \sum_{K=0}^{N-1} A(K)W^{-(n-N)K}$$
$$= \sum_{n=N/2+1}^{N-1} \frac{W^{nK}}{N\Delta t} \sum_{K=0}^{N-1} A(K)W^{-nK} = \sum_{n=N/2+1}^{N-1} X(n)W^{nK}$$

Substituting in the expression for A(K), we obtain

$$A(K) = \sum_{n=-\frac{1}{2}N+1}^{1} X(n)W^{nK} + \sum_{n=0}^{N/2} X(n)W^{nK}$$
$$= \sum_{n=\frac{1}{2}N+1}^{N-1} X(n)W^{nK} + \sum_{n=0}^{N/2} X(n)W^{nK}$$
$$= \sum_{n=0}^{N-1} X(n)W^{nK}$$

Thus
$$A(K) = \sum_{n=0}^{N-1} X(n)W^{nK} \quad \& \quad X(n) = \frac{1}{N\Delta t} \sum_{K=0}^{N-1} A(K)W^{-nK}$$

are Fourier transform pair for sampled periodic band limited functions.

APPENDIX-C

$$X(J) = X'(J) + jX''(J) = \sum_{K=0}^{N-1} A(K)W^{JK}$$

$$= \sum_{K=0}^{N-1} A'(K)W^{JK} + j\sum_{K=0}^{N-1} A''(K)W^{JK}$$

$$A(K) = \sum_{J=0}^{N-1} X(J)W^{-JK} = \sum_{J=0}^{N-1} X'(J)W^{-JK} + j\sum_{J=0}^{N-1} X''(J)W^{-JK}$$

$$A(N-K) = \sum_{J=0}^{N-1} X(J)W^{-J(N-K)} = \sum_{J=0}^{N-1} X(J)W^{-JN}.W^{-JK}$$

$$= \sum_{J=0}^{N-1} X(J)W^{JK}$$

$$= \sum_{J=0}^{N-1} X'(J)W^{JK} + j\sum_{J=0}^{N-1} X''(J)W^{JK}$$

because X'(J) & X"(J) are real.

$$A^*(N-K) = \sum_{J=0}^{N-1} X'(J)W^{-JK} - j\sum_{J=0}^{N-1} X''(J)W^{-JK}$$

$$\tfrac{1}{2}\left[A(K) + A^*(N-K)\right] = \sum_{J=0}^{N-1} X'(J)W^{-JK} = A'(K)$$

$$\tfrac{1}{2}\left[A(K) - A^*(N-K)\right] = j\sum_{J=0}^{N-1} X''(J)W^{-JK} = jA''(K)$$

i.e.

$$A'(K) = \tfrac{1}{2}\left[A(K) + A^*(N-K)\right]$$
$$A''(K) = \frac{1}{2j}\left[A(K) - A^*(N-K)\right]$$

APPENDIX-D

$$X(J)=\sum_{K=0}^{N-1} A(K)W^{JK} = \sum_{K=0}^{\frac{1}{2}N-1} A(K)W^{JK}+ \sum_{K=N/2}^{N-1} A(K)W^{JK}$$

Letting

$m=K-\frac{1}{2}N$

$$X(J)= \sum_{K=0}^{N/2-1} A(K)W^{JK}+ \sum_{m=K=0}^{N/2-1} A(\tfrac{1}{2}N+m)W^{J(\frac{1}{2}N+m)}$$

$$= \sum_{K=0}^{N/2-1} A(K)W^{JK}+(-i)^J \sum_{m=0}^{N/2-1} A(\tfrac{1}{2}N+m)W^{Jm}$$

$$X(2r)= \sum_{K=0}^{N/2-1} A(K)(W^2)^{rK}+ \sum_{m=0}^{N/2-1} A(\tfrac{1}{2}N+m)(W^2)^{rm}$$

$$= \sum_{K=0}^{N/2-1} \left[A(K)+A(\tfrac{1}{2}N+K)\right](W^2)^{rK}= \sum_{K=0}^{N/2-1} A'(K)(W^2)^{rk}$$

similarly,

$$X(2r+1)= \sum_{K=0}^{N/2-1} W^K\left[A(K)-A(\tfrac{1}{2}N+K)\right](W^2)^{rK}= \sum_{K=0}^{N/2-1} A''(K)(W^2)^{rK}$$

Compairing,we obtain:

$$A'(K)=A(K)+A(\tfrac{1}{2}N+K)$$

$$A''(K)=\left[A(K)-A(\tfrac{1}{2}N+K)\right]W^K$$

or

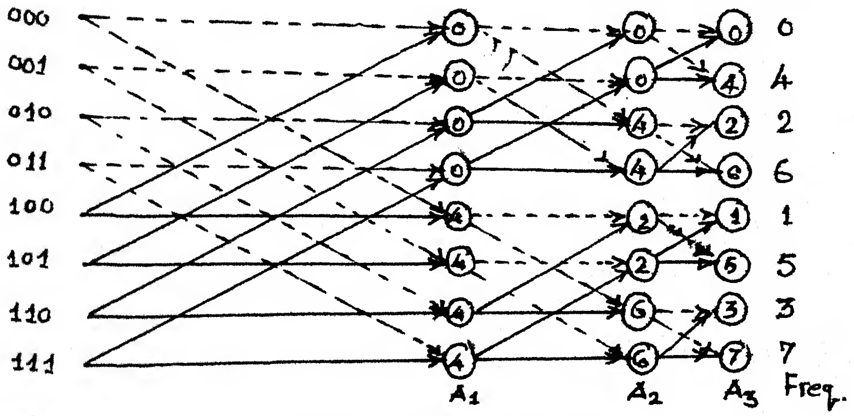
$$A(K)=\tfrac{1}{2}\left[A'(K)+A''(K)W^{-K}\right]$$

$$A(\tfrac{1}{2}N+K)=\tfrac{1}{2}\left[A'(K)-A''(K)W^{-K}\right].$$

APPENDIX-E

TREE-GRAPH REPRESENTATION OF FACTORED MATRICES

The generalized factored matrices can be obtained with the help of tree-graph representation. Taking as an example, N=8, the factored matrices already developed in sec.5 can be represented as a tree-graph as shown below:



Tree-graph for N=8

The sampled data A(K) are represented by the Oth vertical column of nodes on the left of the graph. The ist vertical column of nodes corresponds to vector $A_i(K)$, the 2nd column corresponds to $A_2(K)$ and the last column corresponds to $A_3(K)=X(\bar{K})$. Each node is entered by a dashed and a solid line and within each node is an integer. The solid line brings a quantity from one of the previous nodes, multiplies the quantity by W^p, where p is the integer in the circle, and the product is added to the quantity brought by the dashed line. For example, at node $A_i(ooo)$,

$$A_i(ooo)=A(ooo)+W^o.A(ioo)$$

A generalized method of establishing a graph of the form shown above for any $N=2^M$ has been developed by Brigham and Morrow[2]. The generalized graph is formulated by the following set of rules:

I. Assume N sample values of the time function A(t) constitute the first vertical array of nodes, denoted by A(K) where K=o,i,2,.....,N-i represent the sample times or address (locations) of of the function A(t);argument K is expressed as a binary number.This binary number determines the location of node A(K).

The other arrays, $A_L(K)$, are drawn successively to the right,and the nodes in each array are addressed as binary numbers as mentioned above.At each node,a number is written as explained by rule 2.

2. The number in the circle of the K<u>th</u> node in the L<u>th</u> array is found by:

(a) writing the number K in binary

(b) scaling or sliding this number M-L places to the filling in the newly opened bit positions on the left by zeros.

(c) reversing the order of bits

For example,refering to the graph for N=8,the node 4 in column 2 has the number determined below:

Node 4 expressed in binary	:	ioo
Number of right shifts required	:	3-2=i
Binary number shifted one bit to the right & filled by zeros in the blanks created on the left	:	oio
Binary number after bit inversion	:	oio=2

Similary,the 3rd node in the 3rd column has the following number in it:

Node 3 expressed in binary	:	oii
Number of right shifts required	:	3-3=o
Binary number after shifting	:	oii
Binary number after bit inversion	:	iio=6

3. In the Lth array, node K (in binary form $K_{M-i},\ldots,K_o$) has a solid line drawn to it from a node in the L-ith array. The address of the node in the L-ith array is the same as node K, except that bit K_{M-L} must be a one. The dashed line comes from a node in the L-ith array whose address is the same, but bit K_{M-L} must be a zero.
For example, let

M=3, L=2, and K=3=(oii)

The solid line comes from ooi=3rd node of the ist column.

The dashed line comes from ooi=ist node of the ist column.

4. The Fourier transform values are found in the last array, the location or argument of these transform values is flipped in binary order. For example, the transform component located in ooi is actually the frequency component ioo, which is the bit reversal of the argument ooi. The address of the last array must be bit reversed before the final transform values are obtained.

The tree-graph merely corresponds to the factoring of W^{JK}. From Eq.( ), W^{JK} can be factored as:

$$W^{JK}=W^{JK \bmod N}$$

$$=W^{K_{M-i}(J_o,o\ldots,o)}\cdot W^{K_{M-2}(J_i,J_o,o\ldots,o)}\ldots\ldots\ldots$$

$$W^{K_{M-L}(J_{L-i},\ldots,J_o)}\ldots\ldots W^{K_o(J_{M-i},\ldots\ldots J_o)}$$

For M=3

$$W^{JK}=W^{JK \bmod 8}$$

$$=W^{K_2(J_o,o,o)}\cdot W^{K_i(J_i,J_o,o)}\cdot W^{K_o(J_2,J_i,J_o)}$$

In the tree-graph the node A(K), the Kth time sample, is connected to any of the spectrum nodes $A_3(K)$ by M lines.

For example, A(ooo) & A_3(ooo) are connected only by three dashed lines; these M lines correspond to factoring of W^{JK}. The solid or dashed characteristic of the lines depend on the value of K_L. If K_L=I, the line is solid; for K_L=0, it is dashed.

The Fourier transform values with the node J_2,J_i,J_o end up in position J_o,J_i,J_2. Starting in the 0th array at position K_2,K_i,K_o, proceed by the solid and dashed lines to location J_o,J_i,J_2 to get the contribution of node K_2,K_i,K_o to the transformed nodě J_2,J_i,J_o. For example, start at locati location oii and head for ooi to get the contribution of A(o A(oii) to the Fourier component A_3(ioo).

Array	Position	Power of W
0	K_2,K_i,K_o(oii)	
I	J_o,K_i,K_o(oii)	J_o,0,C(o[illegible])=0
2	J_o,J_i,K_o(ooi)	J_i,J_o,0(ooo)=0
3	J_o,J_i,J_2(ooi)	J_2,J_i,J_o(ioo)=4

```
$JOBELG008,NAME  K B OHRI            ELECTRICAL
$IBJOB
$IBFTC          NODECK
      DIMENSION  AR(64),AI(64),FR(64),FI(64),C(64),AMP(64),DIFF(64
      DIMENSION  PP(6),Q(6),QQ(6),R(6),RR(64),RI(64)
      READ1,N1,N2
1     FORMAT(2I5)
C     MAKE N1 THE LARGER INTEGER
700   IF(N1-N2)701,702,702
701   NTEMP=N1
      N1=N2
      N2=NTEMP
702   N=N1*N2
      PAI=3.14159
      XN1=N1
      XN2=N2
      XN=N
      XM1=0.1+ALOG(XN1)/ALOG(2.)
      XM2=0.1+ALOG(XN2)/ALOG(2.)
      M1=XM1
      M2=XM2
      PRINT1,N1,N2
      PRINT1,M1,M2
C     DATA GEN. HAS STARTED
      DO2I=1,N
      AR(I)=1.
      AI(I)=0.
      XI=I-1
      RR(I)=10.*COS(XI*PAI/16.)
 2    RI(I)=10.*SIN(XI*PAI/16.)
      PRINT1100
 1100 FORMAT(37X,43HDATA FOR ONE DIMENSIONAL HOLOGRAM SYNTHESIS)
      PRINT505,(AR(I),I=1,N)
      DO811I=1,N
      D=(AR(I))**2+(AI(I))**2
 811  C(I)=SQRT(D)
C     DATA GEN. IS OVER
      LCHECK=1
  600 ISC=0
C     J IS BEING USE4 WITH N2 WHILE L IS BEING USED WITH N1
      J=0
      L=1
      ML1=M1-L
  310 J2=J-1
      MJ2=M2-J
      ZZ=1.
      SS2=0.
C     TO TAKE CARE O6 ONE DIMENSIONAL CASE
 309  IF(M2)308,308,307
 308  S=0.
      SS2=0.
      Z1=1.
      ZZ=1.
      GOTO400
```

```
307    IF(J2-1)301,302,302
  302 DO303I=1,J2
  303 QQ(I)=0.
  304 S2=0.
      SS2=0.
      ZZ=1.
      DO305I=1,J2
      K=I-1
      ZZ=ZZ*QQ(I)
      S2=S2+(2.**K)*QQ(I)
      IIN=M2-I
  305 SS2=SS2+(2.**IIN)*QQ(I)
      P=2.*PAI*S2*(2.**MJ2)/XN2
      IF(MJ2-1)306,301,301
  306 S=0.
      GOTO400
C     TO TAKE CARE OF ONE DIMENSIONAL CASE
301   IF(M2)309,309,401
401   DO3I=1,MJ2
    3 PP(I)=0.
  300 S=0.
      Z1=1.
      DO4I=1,MJ2
      K=I-1
      Z1=Z1*PP(I)
    4 S=S+(2.**K)*PP(I)
  400 Z2=1.
```

```
    5 L1=L-1
      IF(L1-1)60,6,6
   60 SS=0.
      GOTO9
    6 ML1=M1-L
      DO7I=1,L1
    7 Q(I)=0.
   70 S1=0.
      SS=0.
      Z2=1.
      DO8I=1,L1
      K=I-1
      Z2=Z2*Q(I)
      S1=S1+(2.**K)*Q(I)
      IIN=M1-I
    8 SS=SS+(2.**IIN)*Q(I)
      IF(J-1)80,81,81
80    P=2.*PAI*S1*(2.**ML1)/XN1
   81 Z3=1.
      IF(ML1-1)90,9,9
   90 SSS=0.
      GOTO12
    9 DO10I=1,ML1
   10 R(I)=0.
  100 Z3=1.
      SSS=0.
      DO11I=1,ML1
      K=I-1
      Z3=Z3*R(I)
   11 SSS=SSS+(2.**K)*R(I)
   12 SSSS=(S+SS2)*XN1+SS+SSS+1.
      IF(J-1)120,121,121
  120 T=SSSS+(2.**ML1)
      IS=SSSS
      IT=T
C     PRINT500,IS,IT,L
C500  FORMAT(3(5X,I5))
      IF(L1-1)13,14,14
13    P=0.
14    IF(LCHECK-2)141,142,142
  141 F=SIN(P)
      SCALE=1.
      GOTO143
  142 F=-SIN(P)
      SCALE=0.5
143   CALCRS=AR(IS)+1R(IT)*COS(P)-AI(IT)*F
      CALCRS=CALCRS*SCALE
      CALCIS=AI(IS)+1I(IT)*COS(P)+AR(IT)*F
      CALCIS=CALCIS*SCALE
      CALCRT=AR(IS)-1R(IT)*COS(P)+AI(IT)*F
      CALCRT=CALCRT*SCALE
      CALCIT=AI(IS)-1I(IT)*COS(P)-AR(IT)*F
      CALCIT=CALCIT*SCALE
      AR(IS)=CALCRS
      AI(IS)=CALCIS
      AR(IT)=CALCRT
      AI(IT)=CALCIT
      IF(L-M1)150,151,151
C151  PRINT501,XR(IS,M1),XI(IS,M1),XR(IT,M1),XI(IT,M1)
C501  FORMAT(4(7X,F13.8))
C     TO TAKE CARE OF ONE DIMENSIONAL CASE
  121 T=SSSS+(2.**MJ2)*XN1
      IS=SSSS
      IT=T
C     PRINT502,IS,IT,J
      IF(J2-1)130,140,140
130   P=C.
140   IF(LCHECK-2)144,145,145
  144 F=SIN(P)
151   IF(M2)800,800,150
      SCALE=1.
      GOTO146
  145 F=-SIN(P)
      SCALE=0.5
146   CALCRS=AR(IS)+AR(IT)*COS(P)-AI(IT)*F
      CALCRS=CALCRS*SCALE
      CALCIS=AI(IS)+AI(IT)*COS(P)+AR(IT)*F
      CALCIS=CALCIS*SCALE
      CALCRT=AR(IS)-AR(IT)*COS(P)+AI(IT)*F
      CALCRT=CALCRT*SCALE
      CALCIT=AI(IS)-1I(IT)*COS(P)-AR(IT)*F
      CALCIT=CALCIT*SCALE
      AR(IS)=CALCRS
      AI(IS)=CALCIS
      AR(IT)=CALCRT
      AI(IT)=CALCIT
      IF(J-M2)15,152,152
  152 ISC=ISC+1
      ITC=ISC+N/2
      FR(ISC)=CALCRS
      FI(ISC)=CALCIS
      FR(ITC)=CALCRT
      FI(ITC)=CALCIT
C     PRINT504,FR(ISC),FI(ISC),FR(ITC),FI(ITC)
      GOTO15
C     TO TAKE CARE OF ONE DIMENSIONAL CASE
800   ISC=ISC+1
      ITC=ISC+N/2
      FR(ISC)=CALCRS
      FI(ISC)=CALCIS
      FR(ITC)=CALCRT
      FI(ITC)=CALCIT
  150 IF(Z3-1.)17,15,15
   15 IF(Z2-1.)20,16,16
   16 IF(Z1-1.)23,26,26
   26 IF(Z2-1.)27,30,30
   17 I=1
```

```
18 R(I)=R(I)+1.
   IF(R(I)-2.)100,19,19
19 R(I)=0.
   I=I+1
   GOTO18
20 I=1
21 Q(I)=Q(I)+1.
   IF(Q(I)-2.)70,22,22
22 Q(I)=0.
   I=I+1
   GOTO21
23 I=1
24 PP(I)=PP(I)+1.
   IF(PP(I)-2.)300,25,25
25 PP(I)=0.
   I=I+1
   GOTO24
27 I=1
28 QQ(I)=QQ(I)+1.
   IF(QQ(I)-2.)304,29,29
29 QQ(I)=0.
   I=I+1
   GOTO28
30 L=L+1
   IF(L-M1)301,301,31
31 L=M1+1
   J=J+1
```

```
      IF(J-M2)310,310,32
32    LCHECK=LCHECK+1
      IF(LCHECK-3)33,35,35
33    DO34I=1,N
      D=FR(I)**2+FI(9)**2
      AMP(I)=SQRT(D)
C     MODIFICATION FOR HOLOGRAPHY
      AR(I)=AR(I)+RR(I)
      AI(I)=AI(I)+RI(I)
      AR(I)=AR(I)**2+AI(I)**2
34    AI(I)=0.
C     MODIFICATION FOR HOLOGRAPHY
      PRINT1000
504   FORMAT(20X,4(7X,F13.8))
1000  FORMAT(46X,34H6OURIER TRANSFORM IS BEING WRITTEN)
      PRINT505,(FR(I),FI(I),AMP(I),I=1,N)
505   FORMAT(6(7X,F13.8))
      GOTO600
35    XERROR=0.
      DO36I=1,N
      D=FR(I)**2+FI(I)**2
      AMP(I)=SQRT(D)
      DIFF(I)=C(I)-AMP(I)
36    XERROR=XERROR+(DIFF(I))**2
      XERROR=SQRT(XERROR/XN)
      PRINT1001
1001  FORMAT(50X,28H8OLOGRAM IS BEING REPRODUCED)
```

```
      PRINT505,(FR(I),FI(I),AMP(I),I=1,N)
      PRINT1002
1002  FORMAT(47X,33HTHE R.M.S ERROR IN THE PROGRAM IS)
      PRINT37,XERROR
37    FORMAT(55X,F13.8)
  704 STOP
      END
$ENTRY
   32    1
```